



Apple IIGS

#98: Aren't Windows A Pane?

Revised by: Dave Lyons
Written by: Dave Lyons

May 1992
January 1991

This Technical Note describes interesting Window Manager things.

Changes since January 1991: Noted that in System 6.0 it's safe to use Window color table resources. Added a section on changing the desktop pattern or picture.

Changing the Desktop Pattern or Picture

The best way to set a new desktop pattern or picture is as follows. This works with the Finder and other desktop applications.

1. Use `MessageCenter` to delete message 2, the desktop message. (If there wasn't one, that's fine—there still isn't.)
2. Use `MessageCenter` to create a new message 2, containing the pattern or picture you want (see the Window Manager chapter of *Apple IIGS Toolbox Reference*, Volume 3).
3. Call `Desktop` (in the Window Manager) with a `deskTopOp` of 8 and a `dtParam` of \$00000000. This notifies any part of the system that cares (such as the Finder) that there is a new desktop pattern.
4. Call `Desktop` with a `deskTopOp` of 4 and a `dtParam` of \$00000000 and keep the result.
5. Call `Desktop` with a `deskTopOp` of 5 and use the result from step 4 as `dtParam`. This sets the desktop pattern to what it already is, forcing the desktop to redraw (this works whether a pattern, picture, or pointer to desktop-drawing routine is involved).

A Warning About Window Color Table Handles And Resources

The System 6.0 Window Manager fixes the problem described below. If your application requires System 6, you can safely ignore this section.

All versions of the Window Manager that support window color tables specified as handles or resources, up to and including System Software 5.0.4, work unreliably when a standard window's color table is supplied by handle or resource ID.

The problem is not immediately obvious; only one bit of memory is accidentally cleared, but the address is unpredictable in advance. (When unlocking the color table handle, the standard window definition procedure attempts to unlock the handle manually by turning off bit 15 of word offset +4 in the master pointer record. But it gets the high and low words of the handle reversed and usually turns off bit 15 of the word at offset \$80E4 or \$80E5 in some bank of RAM determined by the low byte of the handle.)

The solution is to avoid supplying color table handles or resource IDs to the Window Manager. Supply color table pointers instead. You can get a color table pointer from a color table resource ID by calling `LoadResource` on the color table resource, locking the handle and dereferencing it. Memory is less fragmented if color table resources used in this way are marked as `attrFixed`.

One method is to put the window color table pointer into the window template before calling `NewWindow2`. If you are creating the window from an `rWindParam1` resource, you need to use `LoadResource` to get the template into RAM so that you can stuff the color table pointer into the template. (Be sure to change the `moreFlags` field to indicate that the color table is a pointer, if the template indicates it's a resource.) After you create the window with `NewWindow2` (by handle), use `ReleaseResource` to release the `rWindParam1` resource.

Another method is to create the window as invisible and pass the window color table pointer to `SetFrameColor` before calling `ShowWindow`.

Further Reference

- *Apple II GS Toolbox Reference*, Volumes 2-3